MINSTREL 4TE

A488

IEEE-488 Controller

Instruction Manual

USER'S MANUAL

A488

IEEE-488 Controller

Instruction Manual

SSM MICROCOMPUTER PRODUCTS, INC.
2190 Paragon Drive
San Jose, California 95131

TELEPHONE: (408) 946-7400
TWX: 910-338-2077
TELEX: 171171
System Sonic (408) 946-8191

™A488 is a trademark of SSM Microcomputer Products, Inc.

Written by Malcolm T. Wright
Edited by Dan Fischler
Illustrated by Judith A. Sisko

©1981 SSM Microcomputer Products, Inc.
All Rights Reserved

Part No. MN0004
Revision 1
April, 1983

TABLE OF CONTENTS

1.0 INTRODUCTION

2.0 GENERAL DESCRIPTION OF IEEE 488

- 2.1 Type of Device
- 2.2 IEEE 488 Commands
- 2.3 Signal Lines
- 2.4 Timing

3.0 HOW TO USE THE A488

- 3.1 Installation
- 3.2 Simple Checkout
- 3.3 Software (General)
- 3.4 Software (Unique Requirements)
  - 3.4.1 Slot
  - 3.4.2 Variables
  - 3.4.3 Return Response
  - 3.4.4 Symbol Table
  - 3.4.5 Service Request
  - 3.4.6 Sound
  - 3.4.7 Pause
  - 3.4.8 End-Of-String (EOS)
  - 3.4.9 Switch Entry Point
  - 3.4.10 Bus Timeout
- 3.5 Software Commands
- 3.6 Error Messages
- 3.7 RAM Option
- 3.8 Entry Point Map
- 3.9 Software Example

4.0 THEORY OF OPERATION

- 4.1 ROM Selection
- 4.2 ROM/RAM Circuitry
- 4.3 Device Select Addressing
- 4.4 IFC/REN/ATN Signals
- 4.5 IEEE 488 Buffering

5.0 TROUBLESHOOTING HINTS

- 5.1 Bus Management Lines ATN, REN & IFC
- 5.2 ROM Mapping Enable Circuitry
- 5.3 ROM Mapping Disable Circuitry
- 5.4 MC68488 Chip Function

6.0 WARRANTY

APPENDIX:

Schematic (INSERT)
Parts List
MC68488 Specification Sheet
MC3446A Specification Sheet

1.0 INTRODUCTION

The SSM A488 board provides a powerful communication interface between an Apple II™ and over 1200 digitally controlled products for test, measurement and control. The A488 supports the instrument interface bus that was developed by Hewlett-Packard (HP-IB)™ and is now sanctioned by the Institute of Electrical and Electronic Engineers (IEEE-488 1978), American National Standards Institute (ANSI MC1.1), and International Electrotechnical Commission (IEC 625-1). The 488 bus interface is used by over 150 manufacturers of digitally controlled equipment and is used in over 14 countries throughout the world.

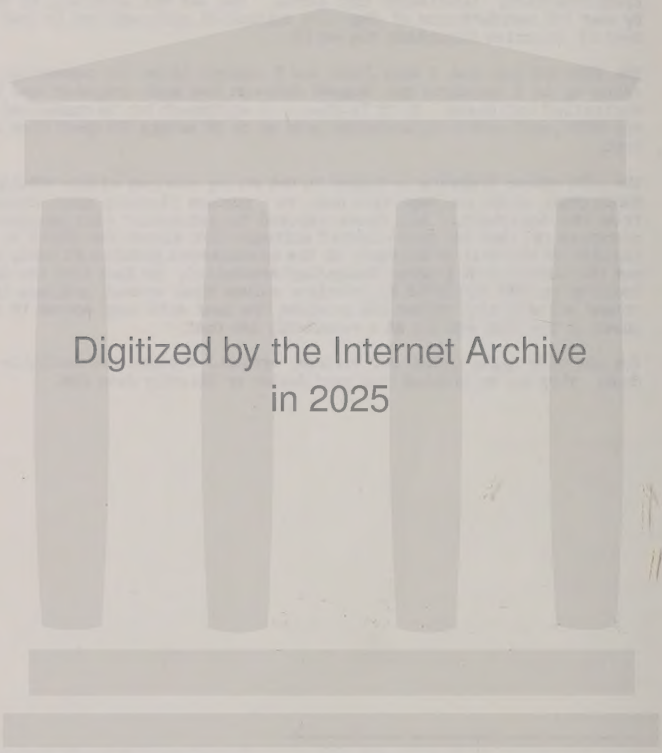
The IEEE 488 bus uses 8 data lines and 8 control lines for communication rates up to 1 megabyte per second between the host computer and the controlled equipment. Up to 14 pieces of equipment can be connected to the A488 board with a transmission path up to 20 meters (66 feet) from the host.

The A488 driver firmware is linked to the string routines within APPLESOFT BASIC ONLY, since the user will want to process floating point numbers from the equipment. All those exposed to automatic test equipment recognize the need for user-oriented software that allows the effort to be focused on the unit under test, or the measurement problem at hand, and not the mechanics of program design/implementation. We feel that the A488 board as an IEEE 488/APPLE II interface solves these unusual problems in a unique and effective manner and provides the user with easy access to the power of the IEEE 488 bus at a remarkably low cost.

For use with Apple PASCAL and Fortran, optional drivers are available on disk. They may be ordered from your dealer or directly from SSM.

Apple II, Apple II Plus, and Applesoft are trademarks of APPLE COMPUTER, INC., 10455 Bandley Drive, Cupertino, CA.

HP-IB is a trademark of HEWLETT-PACKARD, 1501 Page Mill Road, Palo Alto, CA.

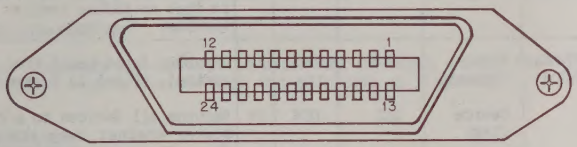


Digitized by the Internet Archive
in 2025

https://archive.org/details/manualsplus_10282

2.0 GENERAL DESCRIPTION OF IEEE 488

The IEEE 488 instrument interface bus employs 8 bi-directional data lines and 8 control lines to form a parallel communication interface between the controller and the equipment. Three of the 8 control lines (NRFD, DAV, NDAC) are used for handshaking between the controller (host) and an instrument when data is sent. The interfacing connector is a 24 pin connector that provides eight additional ground connections to allow for twisted pairs on the 8 control lines to reduce noise coupling, if required. Fifteen pieces of equipment (1 host and 14 slaves) can be placed on the IEEE 488 bus at one time, with a cable length up to 20 meters.



Pin 1	DIO1	Pin 13	DIO5
Pin 2	DIO2	Pin 14	DIO6
Pin 3	DIO3	Pin 15	DIO7
Pin 4	DIO4	Pin 16	DIO8
Pin 5	EOI	Pin 17	REN
Pin 6	DAV	Pin 18	DAV/GND
Pin 7	NRFD	Pin 19	NRFD/GND
Pin 8	NDAC	Pin 20	NDAC/GND
Pin 9	IFC	Pin 21	IFC/GND
Pin 10	SRQ	Pin 22	SRQ/GND
Pin 11	ATN	Pin 23	ATN/GND
Pin 12	SHIELD	Pin 24	SIGNAL GROUND

2.1 TYPE OF DEVICE

A device equipped with an IEEE 488 interface can perform one or more of the following functions:

CONTROLLER	A device capable of controlling an instrument over the IEEE bus by addressing and sending commands. The Apple II with the A488 board becomes a controller.
TALKER	A device capable of sending data to other instruments, e.g., a digital voltmeter outputting data, counter outputting data or a paper tape reader. (Not available with A488.)
LISTENER	A device capable of receiving data from other instruments, e.g., a line printer, programmable signal generator, programmable power supply, or a paper tape punch. (Not available with A488.)

2.2 IEEE 488 COMMANDS

Data that is transmitted over the IEEE 488 bus is normally a 7-bit ASCII (American Standard Code for Information Interchange) code with the eighth

bit available for parity checking, if needed. When a command is transmitted, certain letters from the ASCII table have been given special meaning, as follows:

(ATN line is held true [low] for the command mode.)

TYPE	COMMAND	MNEMONIC	ASCII	HEX	COMMENTS
UNADDRESS COMMANDS	Unlisten	UNL	?	3F	Clears bus of all listeners.
	Untalk	UNT		5F	Unaddresses the current talker so that no talker remains on the bus.
UNIVERSAL COMMANDS	Local Lockout	LLO	DC1	11	Disables front panel from user's control. Panel is locked out.
	Device Clear	DCL	DC4	14	Returns all devices to a cleared state, whether they are addressed or not.
	Serial Poll Enable	SPE	CAN	18	Enable Serial Poll mode to read the device's status.
	Serial Poll Disable	SPD	EM	19	Disable Serial Poll mode.
	Parallel Poll Unconfigure	PPU	NAK	15	Set all devices on the bus with parallel poll capability to a pre-defined condition.
ADDRESSED COMMANDS	Go to Local	GTL	SOH	01	Enable instrument front panel to allow manual user control.
	Selective Device Clear	SDC	EOT	04	Clears only one device on the bus.
	Group Execute Trigger	GET	BS	08	Start all pre-setup equipment at the same time.
	Parallel Poll Configure	PPC	ENQ	05	This command permits the data lines to be assigned to a device deviceonthebusfora response to a "PPU". (Not available with A488).
	Take Control	TCT	HT	09	Pass control from the preset controller to another. (Not available with A488.)

UNADDRESS COMMANDS

Unlisten: This command unaddresses all listeners that have been previously addressed to listen.

Untalk: This command unaddresses any talker that had been previously addressed to talk.

UNIVERSAL COMMANDS

Bus devices capable of responding to these commands from the controller will do so at any time regardless of whether or not they are addressed.

ADDRESSED COMMANDS

These commands are similar to universal commands except that they are recognized only by devices that are addressed as listeners.

2.3 SIGNAL LINES

There are 16 signal lines used on the IEEE 488 bus for communication. Negative logic levels are used on these lines, so True = Low, False = High.

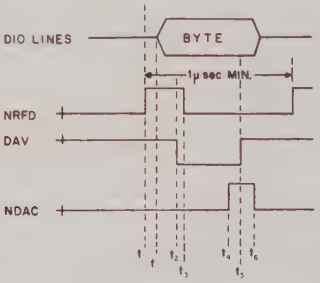
NAME	PIN	FUNCTION
DIO 1	1	Bi-directional data line, LSB
2	2	" " "
3	3	" " "
4	4	" " "
5	13	" " "
6	14	" " "
7	15	" " "
8	16	Bi-directional data line, MSB
NRFD	7	Not Ready For Data. NRFD line will be high when the listener is ready to receive data.
DAV	6	DAta Valid. Goes low when data is stable and NRFD is high.
NDAC	8	Not Data ACcepted. This signal stays low after DAV until the listener has accepted the data and then goes high until DAV goes false.
IFC	9	InterFace Clear. Only a controller can activate this line. When this signal is set true, all talkers and listeners go to their inactive states.
ATN	11	ATteNtion. Goes low when a command is being sent. This line is high when instrument data is being transmitted.
REN	17	Remote ENable. The controller sets this signal low and then addresses the devices to listen before they will operate under remote control.

NAME	PIN	FUNCTION
EOI	5	End Or Identify. This line is used to indicate the end of a multiple byte transfer or, in conjunction with ATN, to execute a parallel polling sequence.
SRQ	10	Service ReQuest. This line goes low when a device is requesting service from the controller.

2.4 TIMING

Three handshaking lines are used for sending data: 1) NRFD, 2) DAV, and 3) NDAC. The sequence of the signals on these lines is critical for correct IEEE 488 operation. The rate of data transfer is limited to the slowest device that is participating in the transfer. (Data rates up to 500 kilobytes per second are not uncommon.)

Low level = True
High level = False



t_0 At this time all listeners allow NRFD to go high when they are ready for more data. NDAC is held low by the listener.

t_1 The controller senses that NRFD is high and NDAC is low and now places its next byte of data on the DIO lines.

t_2 The controller then drops DAV low when it knows the data will be valid.

t_3 The first listener to receive the data drops NRFD low, then all other listeners follow.

t_4 When all listeners have received the data, NDAC is raised high.

t_5 The controller senses that NDAC is high and then removes the DAV signal (set high). One byte of data has been transferred. Shortly after this, data can change states on the DIO lines.

t_6 When DAV is set high, then NDAC is pulled low by the listener to get ready for another byte of data.

3.0 HOW TO USE THE A488

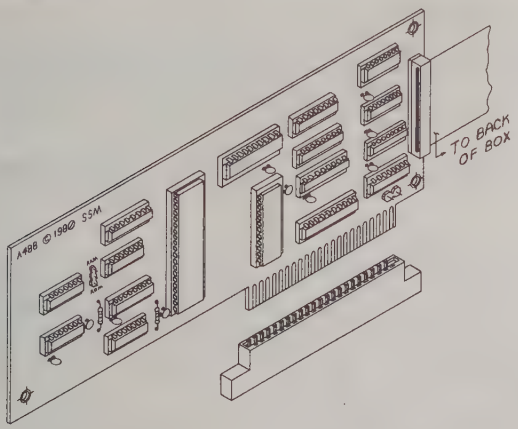
3.1 INSTALLATION

The A488 board is designed to plug into one of the slot connectors inside the Apple II, usually slots 1 thru 5. Any one of the 8 slot connectors can be picked except slot 0, which is used for future Apple bus expansion. (Remember that slots 6 & 7 are normally reserved for use by the Apple disk drives, so you will probably not want the board there.)

Make sure that the ROM/RAM option on the A488 board has the mini-jumper on the ROM side (lower 2 pins of the 3 pin plug) to enable the on-board ROM firmware. Looking at the component side of the A488, install the 26 pin socket on the end of the ribbon cable to the right edge of the board with the cable going to the right.

CAUTION: Damage can result if the cable is installed incorrectly.

Turn the power off on the Apple II. Install the A488 in the desired slot with the cable going out one of the openings provided in the back of the Apple. With the A488 firmly installed, you may now turn the power on the Apple computer.



3.2 SIMPLE CHECKOUT

You can do a simple checkout that will prove if you can read the A488 firmware. More detailed tests are available in Section 5.0.

First, enter the Apple II's monitor routine. If you have an asterisk character (prompt character) on the left side of the screen, you are already in the monitor. If you have a "]" character, type:

```
<CR> = carriage return
CALL -151 <CR>      Entry point to monitor
```

Now type the following monitor command:

```
N = slot number, 1 thru 7
CN00L <CR>
```

This command will disassemble the first 20 lines of the A488's firmware code. The first three lines should read as follows:

```
STA   CFFF
LDA   7F8
PHA
```

To restore the system, press the reset button on the Apple II, or turn power off and then power up the Apple.

3.3 SOFTWARE (GENERAL)

The A488 firmware is set up for APPLESOFT ONLY since floating point math will become important in making a measurement or analyzing incoming IEEE 488 data. All communication on the IEEE 488 bus is done with ASCII characters so the string manipulation ability of Applesoft makes communication easy.

Controlling IEEE 488 devices through Applesoft has been made as simple as possible. All devices can be commanded to do something by their name, which can be up to 16 characters in length; for example:

```
DVM1
COUNTER
SIGNALGENERATOR
```

Sending function setup instructions to a device is simply a matter of putting the instruction string into a string called FS$. To tell the A488 where to send a command instruction, a command letter plus the name of the device is placed in a string called CI$. Responses from the equipment back to the Apple II will be put into a string called RS$ for the user to process.

There are four easy steps that must be at the beginning of your BASIC program to set up needed string variables, the A488 firmware and the IEEE 488 bus.

1. Define which slot your A488 is plugged into. There must not be any other variables declared before Step I.

   Example for slot 3:

   ```
   SL%=3
   ```

2. Define your table of IEEE 488 devices by name and address. Also set up string variables for response, function and command.

Example device table with other variables.

```
DT$="COUNTER:13,DVM:05,SIGGEN:01"
RS$=""          (Null string)
FS$=""          (Null string)
CI$=""          (Null string)
PA%=1
```

NOTE: The variables must be assigned EXACTLY in the order shown.

3. Define all software entry points to the A488 firmware.

```
X = 256*SL%          Slot offset value
IC = 49152+X         Generate IFC
SN = IC+16           Normal command entry
FI = IC+32           First entry, Initialize A488
PA = IC+96           Pause entry
SO = IC+192          Sound entry
SR = 49295+SL%*16    Read service request (SRQ)
PE = 1144+SL%        Period of sound (Pitch)
DU = 1272+SL%        Duration of sound
EOS = 1912+SL%       End of string "EOS" delimiter
```

4. Now set up the interface, initialize the software, and define the maximum length of the response string through calculation.

```
CALL FI              Initialize A488
CALL IC              Clear IEEE 488 equipment
FOR T=1 TO 16:RS$=RS$+" ":NEXT
```

NOTE: RS$ must be a minimum of 16 characters. (See Section 3.4.3, Return Response.)

PROGRAM

Now you are ready to write software in BASIC to meet your particular test requirements. An example program to set a frequency counter's function may look like the following:

```
FS$="MODE,RANGE"      "MODE" and "RANGE" would be a string of
                      characters defined by the manufacturer's
                      manual which will set up the counter.

CI$="S,COUNTER"       S = Setup and is followed by the device's
                      name. (Additional commands in Section 3.5.)

CALL SN               Send command to the A488 and send function
                      to the counter.
```

3.4 SOFTWARE (UNIQUE REQUIREMENTS)

3.4.1 Slot

The firmware on the A488 is written to be slot independent, but because of popular usage of certain slots it is recommended that the board be placed into slot 3, 4 or 5.

3.4.2 Variables

In the example software setup in Section 3.3, Steps 1 & 2 define six variables that will be used by the A488 firmware. The firmware drivers **DEMAND** that the first six variables be in the order and type as follows:

```
SL%=3
DT$="DVM:01"
RS$=""
FS$=""
CI$=""
PA%=1
```

The length and address of DT$, RS$, FS$, CI$ and PA% are found by the A488 firmware by using the BASIC symbol table organization.

Each string variable can be up to and including 256 bytes maximum in length. The string DT$, which defines the name and address of each device, can have a device name up to and including 16 characters long. The address after each device name must be two digits preceded by a colon. The address is a Hex value from 00 to 1F and must match the preset address on each piece of equipment.

3.4.3 Return Response

The RS string holds the return response of a device and must be 16 characters long or longer in length. An RS string of insufficient length will result in program termination and an error message. RS$ must be brought to its proper length by calculated means and not through a literal to avoid certain side effects such as the appearance of response data in the literal definition within the user's program. (The BASIC symbol table will point there.) The maximum length of RS$ is 256 characters.

Example: RS$="":FOR I=1 to 16:RS$=RS$+" ":NEXT

Remember that RS$ must be long enough to hold the response data. Sixteen characters is sufficiently long for most instrument responses, but it is the user's responsibility to ensure proper length for those that are longer.

3.4.4 Symbol Table

The A488 firmware uses the Applesoft secondary floating point accumulator for the necessary zero page references ($A5-$AB). The

drivers leave a $00 upon exit. The A488 firmware also takes advantage of the Applesoft symbol table organization to find and use BASIC variables for parameter/result communication. The integer variable format (SL%) is used for more than is readily apparent. The symbol table organizaton for integer variables is as follows:

0 First character (name)
1 Second character (name)
2 High byte (value)
3 Low byte (value)
4 0
5 0
6 0

NOTE: There are three bytes conveniently available for driver usage. Location #4 holds the value of the slot*16. Location #5 holds the value of CI$.

3.4.5 Service Request

The service request line (SRQ) is accessible to the user by reading the appropriate memory location. The line can be read by PEEKing in BASIC.

```
SRQ=49295+16*SLOT
X=PEEK(SRQ)
```

If "X" is equal to 128 or greater, then SRQ is false. If "X" is 127 or less, then SRQ is true. The most significant bit of the byte read by PEEKing is the SRQ flag bit.

No provision has been made to allow the service request line to generate an interrupt, or to initiate any non-user directed action. This technique is consistent with the Applesoft-only approach taken.

3.4.6 Sound

The A488 firmware is set up to generate sound on the Apple II's speaker. Once the board's slot (SL%) is known, two bytes are available to control the sound. The sound routine is useful for prompting the operator in a BASIC test sequence, or sounding an alarm if some measurement is out of range.

```
PE = 1144+SL%
DU = 1272+SL%
```

Location PE can be preset with a number between 1 to 255 which will control the pitch (frequency) of the sound. Location DU can be preset with a number between 1 to 255 which will control the duration of the sound. Calling SO, which is at 49344+256*SL%, will produce the sound.

```
Example: POKE PE,33      For 1000 cycles
         POKE DU,128     For about 1/2 second
         CALL SO
```

PE Number	1	10	33	50	100	200	255	0
Approx. Freq. (Hertz)	15.8K	3.03K	1.0K	666	339	170	133	132

DU Number	128	0
Approx. Duration	0.47 sec	0.95 sec

3.4.7 Pause

At some time you may use a piece of equipment which requires some setup time before you can access it again. The A488 firmware has a built-in timer routine which will allow pause intervals from 1 millisecond to 32 seconds with an optional 65 second pause. The time value is passed as an integer variable labelled PA%. The PA% value equals the time in milliseconds from 1 to 32767 as defined by the user. If PA% is set equal to a negative one, then the time is 65,535 milliseconds.
Routine entry point: PA=49248+256*SL%; SL%=Slot Number

```
Example: Pause for 1000 milliseconds (1 sec.)
         PA%=1000
         CALL PA
```

3.4.8 End-of-String (EOS)

The IEEE 488 standard has evolved over the past 5 years with some redefining of the interface requirements. The 1975 standard for string communication between 488 equipment defined a carriage-return and line-feed as a delimiter. The definition was just unclear enough that some equipment wanted a carriage-return, some a line-feed, some wanted both and some wanted a hardware signal called EOI.

The A488 supports both the 1975 and 1978 IEEE standard with the EOI control signal and a user defined ASCII letter used together as an end-of-string delimiter. Upon A488 initialization, the default ASCII delimiter letter will be set to a line-feed ($0A).

This will support the majority of the recent 488 hardware. If your equipment requires a different delimiter letter, then POKE the decimal value into memory before commanding the device.

```
Example: Set delimiter to a carriage-return.
         Carriage return = 13 decimal

         EOS = 1912+SL%
         POKE EOS,13
```

In version 2.0 of the A488 firmware, EOS now defines the last data character sent as well as the input delimiter. Setting EOS to zero will suppress sending EOS as the last character of a string. EOI true is then required to terminate an input string.

3.4.9 Switch Entry Point

SW is used to point the A488 firmware to another A488 card in another slot in the system. The user should CALL "FI" and "IC" for each card in the system once before using SW.

Example:

```
SL%=4
DT$=""
RS$=""
FS$=""
CI$="":IC=49152+256*SL%
SW=IC+48

D4$="DVM:11,CTR:14"
D3$="SFGEN:10,SPECT:12"

DT$=D4$
CALL FI
CALL IC      INITIALIZE SLOT 4
SL%=3
DT$=D3$
CALL FI
CALL IC      INITIALIZE SLOT 3

SL%=4
DT$=D4$
CALL SW      SWITCH TO SLOT 4

SL%=3
DT$=D3$
CALL SW      SWITCH TO SLOT 3
```

3.4.10 Bus Timeout

The A488 bus timeout is used to avoid "system freeze" when an instrument does not respond to the A488 card. The timeout interval is user-definable in units of milliseconds, with the default being 15 seconds. The user may also override the default by "POKING" a new value as shown:

```
TH=1656+SL%
TL=1528+SL%

POKE TH, (High Byte)
POKE TL, (Low Byte)
```

NOTE: Time = [(High Byte)*256+(Low Byte)] x 10^{-3}
Maximum time equals about 32767 milliseconds.

When an instrument does not respond within the specified interval, the following error message will be displayed:

```
A488 TIMEOUT
SYNTAX ERROR IN LINE XXXX
```

The timeout feature may be disabled by poking zeros onto "TH" and "TL".

3.5 SOFTWARE COMMANDS

The on-board firmware provides a variety of commands to the A488 board.

A488 COMMANDS

S [SETUP]	Sends setup or other one-way information as found in FS$ to the specified listener. **Example:** FS$="MODE,RANGE" CI$="S,COUNTER" CALL SN
M [MEASURE]	Same as "S" but waits for an instrument's response that will be found in RS$ upon completion. **Example:** FS$="D2A" CI$="M,DVM" CALL SN RS$ will hold the DVM's response.
C [DEVICE CLEAR]	Sends a device clear to the specified device. **Example:** CI$="C,SIGGEN" CALL SN Sets the signal generator to standard conditions.
L [GO TO LOCAL]	Sets the specified device to local mode. **Example:** CI$="L,MODMETER" CALL SN The modulation meter is now controllable from the front panel switches.
T [GROUP TRIGGER]	Sends the group trigger command. **Example:** CI$="T,COUNTER" CALL SN Forces all pre-setup equipment to trigger on previous setup ranges, function, mode, etc. Requires a valid device specification.
P [SERIAL POLL]	Performs serial poll on the single device specified. RS$ will contain a one character result. **Example:** CI$="P,COUNTER" CALL SN Status character found in RS$.
R [RESPOND]	Addresses the specified device as a talker and awaits the response string that will be found in RS$. **Example:** CI$="R,POWERMETER" CALL SN RS$ will hold the POWERMETER response.

G [GROUPED RESPONSE]	Allows the Apple II to accept very long string responses (RS$>256) in small packets, if the data is separated by commas. Call the normal entry point "N" times for "N" number of data points. Example: FOR R=1 TO 1024 CI$="G,SPECTANAL" CALL SN CALL PROCESS NEXT R The above example was used on a spectrum analyzer which provided over 1000 points of data. "PROCESS" in this example operates on the data returned in successive commas in the string called RS$. **WARNING:** It is the responsibility of the user to call this command only the number of times EQUAL to the number of data bytes to be returned by the instrument.
O [LOCAL LOCKOUT]	Sets instruments in local lockout mode. Instruments in remote control mode. This command requires a valid device specification. Example: CI$="O,DVM" CALL SN
D [UNIVERSAL CLEAR]	Clears all instruments. This command requires a valid device specification. Example: CI$="D,CTR" CALL SN

A new set of control primitives has been added to the A488 command structure. These new commands allow the user to read and write data directly to the bus so that the user can implement functions not covered in the standard A488 function set.

W [COMMAND WRITE]	The characters found in the string FS$ are written directly to the bus with the ATN line asserted true. No implicit addressing is performed. Secondary addressing, multiple "group execute triggers", etc., may be implemented using this command.
E [DATA WRITE]	The characters found in the string FS$ are written directly to the bus as data bytes with no bus control commands being sent as part of the "E" function. This is a data primitive. "EOS" is appended to the RS$, and "EOI" is asserted true for the last byte sent. No Unlisten is sent upon completion of transfer.

Z [STRING INPUT]	Used in conjunction with the "W" command to accept data from a previously set-up talker, sends no implicit Unlisten, Untalk, or device talk address. The A488 accepts data into the string RS$ until EOS occurs or EOI goes true. No Untalk is sent after end of string. Be sure to define RS$ large enough to hold the block of data to be returned.
I [BINARY OUTPUT]	This is similar to the "X" command except that no ASCII setup data is sent out. The data is stored in binary form as in the "X" command. BW% array is used to store the incoming data until the byte count CT% is satisfied or until EOI is asserted true. Untalk is sent after the transfer is satisfied. BW% is filled starting with position 0, two bytes per array position. **Requires CT% and BW%(N) position dependent variables.**
J [BINARY OUTPUT]	This outputs the contents of the array BW%(N) to the bus. A Listener must have been previously set up using the "W" command. The transfer sends bytes from BW% (word array high, then low) until CT% (byte count) is satisfied. EOI is asserted true on the last byte. Unlisten is sent when the transfer is complete. **Requires CT% and BW%(N) position dependent variables.**

X [PROMPTED BINARY TRANSFER]
 	
Instrument to Controller (Read Device)
	The "X" command requires that two more variables be added to the reserved position list as shown below:
	
	`SL%=4:DT$="CTR:13,SPECT:08"`
	`RS$="":FS$="":CI$=""`
	`PA%=2000`
	`CT%=2048 <==BYTE COUNT`
	`DIM BW%(....) <==BINARY WORD ARRAY`
	
	Note that the BW% array holds WORDS, while CT% is a BYTE COUNT. The "X" function is typically invoked as follows:
	
	`CT%=2048:CI$="X,SPECT"`
	`FS$="LFM,70000,1024" <==PROMPT STRING`
	`CALL SN`
	
	Upon completion BW% array will hold the response in array positions 0-1023.

Q [QUIET CONTROLLER]	The "Q" command allows the specified instrument to talk to other listener devices without data being accepted by the A488 controller. The listening devices must have been set up using the "W" command primitive.
	
	Example:
	`CI$="W,DUMMY"`
	`FS$=CHR$(..)+CHR$(..)+CHR$(..):REMLISTEN ADDRESS`
	`CALL SN`
	`CI$="Q,DATASENDER":REM TALKING DEVICE`
	`CALL SN`
	
	Control will return to the A488 when the talker sends EOI true.

Y [PROMPTED BINARY TRANSFER]
 	
Controller to Instrument (Send to Device)
	The "Y" command requires that two more variables be added to the reserved position list as shown below:
	
	`SL%=4:DT$="CTR:13,SPECT:08"`
	`RS$="":FS$="":CI$=""`
	`PA%=2000`
	`CT%=2048` <===BYTE COUNT
	`DIM BW% (....)` <===BINARY WORD ARRAY
	
	Note that the BW% array holds WORDS, while CT% is a BYTE COUNT. The "Y" function is typically invoked as follows:
	
	`REM:CALCULATE OR READ STORED BW% ARRAY VALUES`
	`CT%=2048:CI$="Y,SPECT"`
	`FS$="WTM,7000,1024"` <===PROMPT STRING
	`CALL SN` <===INVOKE TRANSFER
	
	Upon completion the contents of array BW% have been transferred to device "SPECT". The byte count in CT% determines the number of bytes transferred. The prompt string contents are totally device-dependent but typically specify, action, address, and word count.
	
	NOTE: BW% contents are used or filled with array position zero being used first.

3.6 ERROR MESSAGES

The firmware for the A488 has been set up to check for format errors in the BASIC program. The error messages are as follows:

- **Invalid Command Character.** The first letter in the CI$ string is not one of the command letters listed in Section 3.5.
- **Invalid Command Format.** The comma (,) is missing in the command string CI$.

. **Undefined Device in Command.** The device name in the command string (CI$) does not match any device name in the string table (DT$). See Section 3.3, Step 2, for an example device string.

. **Response String < 16 Characters.** The response string RS$ has been made too small. See Section 3.4.3 for additional information.

. **Device Table Format Error.** The device table DT$ is checked for errors.

1. There must be a colon between the device name and address.
2. The device address must be a Hex number.
3. The device address must be from 00 to 1F.
4. There must be commas separating devices in table.
5. There must be no "spaces" in the device table.

. **Response String Too Short.** The response string RS$ was defined by the user too short to receive all the data from the addressed device. See Section 3.4.3.

3.7 RAM OPTION

If you wish to write your own driver software for the A488, this board has been designed with a RAM memory option. The on-board EPROM (U8) can be replaced with a 2Kx8 RAM such as the Hitachi HM6116P. The 3 pin plug labelled RAM/ROM should have the mini-jumper moved to the RAM position.

The on-board RAM will be bank-switched just like the EPROM was selected. The user must access the slot's I/O space like address C100, C200, C300, etc., to turn on the bank before a machine program can be downloaded into the 2K space starting at C800 Hex. The downloading process may not be possible under Applesoft since the C800 space is constantly turned off. Use the Apple monitor or a machine program to transfer the driver software to the A488 card.

3.8 ENTRY POINT MAP

The four main firmware entry points for FI, IC, SN, SO and PA are shown in the following table. (For reference only.)

HEX ADDRESS (MONITOR Entry)

SLOT NO.	IC	SN	FI	PA	SO
1	C100	C110	C120	C160	C1C0
2	C200	C210	C220	C260	C2C0
3	C300	C310	C320	C360	C3C0
4	C400	C410	C420	C460	C4C0
5	C500	C510	C520	C560	C5C0
6	C600	C610	C620	C660	C6C0
7	C700	C710	C720	C760	C7C0

DECIMAL ADDRESS (BASIC Entry)

SLOT NO.	IC	SN	FI	PA	SO
1	49408	49424	49440	49504	49600
2	49664	49680	49696	49760	49856
3	49920	49936	49952	50016	50112
4	50176	50192	50208	50272	50368
5	50432	50448	50464	50528	50624
6	50688	50704	50720	50784	50880
7	50944	50960	50976	51040	51136

For hardware addresses for the on-board logic functions of the A488, see Section 4.3.

[illegible]

[illegible]

[illegible]

Example of X and Y Transfer Routines

```
5 SL% = 4
6 DT$ = "SFG:11:CTR:13,DVM:05"
7 RS$ = ""
8 FS$ = ""
9 CI$ = ""
10 PA% = 2000
12 CT% = 14
13 DIM BV%(40): REM  DEFINE SIZE OF INTEGER DATA ARRAY
15 IC = 49152 + 256 * SL%
16 FI = IC + 32
18 SN = IC + 16
19 PA = IC + 96
20 SO = IC + 192
35  CALL FI
38  POKE 1142 + SL%,20
39  POKE 1272 + SL%,32
40  CALL IC
50 RS$ = ""
60  FOR T = 1 TO 16:RS$ = RS$ + "": NEXT
300  REM  SEND FUNCTION TO "CTR"
310 CI$ = "Y,CTR":FS$ = "FOR3A+OLA+1.OSO": GOSUB 1000: CALL SN
320 FS$ = "MOH1": CALL SN: REM  SEND FUNCTION TO "CTR"
350 CI$ = "X,CTR":FS$ = "T": CALL SN: REM  READ "CTR"
355  VTAB 12
370  FOR J = 0 TO 7: PRINT BV%(J);"";: NEXT J
380  PRINT
382  GOSUB 5000
390  CALL SO
400  GOTO 350
1000 F$ = MID$(FS$,2) + CHR$ (10):CT% =  LEN (F$)
1020 K = 0:M = 0
1100  FOR J = 1 TO  LEN (F$)
1120  IF K = 0 THEN BV%(M) = ASC ( MID$ (F$,J,1)):K = 1: GOTO 1200
1140 BV%(M) = 256 * BV%(M) + ASC (MID$(F$,J,1))
1145 K = 0:M = M + 1
1200  NEXT J
1220  IF K = 1 THEN BV%(M) = BV%(M) * 256
1300  FOR J = 0 TO M
1340  PRINT BV%(J)
1350  NEXT
1355 FS$ = LEFT$ (FS$,1)
1360  RETURN
5000  FOR J = 0 TO 7
5010 Q =  INT (BV%(J) / 256): PRINT CHR$ (Q)
5020 Q =  INT (BV%(J) - Q * 256): PRINT CHR$ (Q);
5030  NEXT J
5040  PRINT
5050  RETURN
10000  END
```

4.0 THEORY OF OPERATION

4.1 ROM SELECTION

The 2716 EPROM (or HM6116 RAM) on the A488 holds up to 2048 bytes of firmware to run the board. (However, the decoding circuitry on the A488 decodes only 15/16 of this, so the last 128 bytes of the EPROM (or RAM) are not usable. The firmware needs to appear in the 256 byte I/O select memory space for its resident slot address, and when bank switched it will use the reserved expansion ROM space that begins at $C800. The expansion ROM space is designed to be shared by all the slots, and the A488 board uses a protocol defined by Apple Computer to use this space. (See Apple publication for protocol.)

A typical access sequence is as follows:

I/O SLOT READ

A subroutine call from BASIC directs the execution of the processor to the slot address space used by the A488. The Apple pulls the I/O select line (pin 1) low for the duration of the memory access. The upper three address lines (A8, A9 & A10) of the 2716 are made zero, causing only the first 256 bytes of the ROM to be addresed into the I/O space. Three gates from U3 (74LS08) control the logic state of the upper three address lines.

R-S FLIP-FLOP (ON)

The I/O select line also sets an R-S Flip-Flop made up of two gates of U1 (74LS132, pins 8 & 10 to pins 12 & 11). The output of this R-S Flip-Flop is gated by U1, pin 4, which is controlled by U1, pins 1 & 2 during I/O select and I/O strobe. The gated output U1, pin 6 is used to enable the firmware U8 (2716) and the transceiver U10 (74LS245). Once the flip-flop is switched by I/O select going low, then U8 can be accessed in the space $C800 when I/O strobe goes low. The three upper address lines thru U3 also become active, instead of low, during I/O strobe to allow maximum use of the 2K ROM U8.

R-S FLIP-FLOP (OFF)

The R-S Flip-Flop can be reset to free the $C800 expansion area by addressing $CFFF. U4 (74LS138) is used as a simple address decoder for the space from $CF80 to $CFFF. If the Apple accesses any one of the 128 bytes under $CFFF, a reset signal is applied to U1, pin 9. Once reset has been applied to the R-S flip-flop, addressing the expansion area will not enable the A488 firmware (U8) unless the A488's slot is addressed again.

4.2 ROM/RAM CIRCUITRY

The ROM/RAM selection circuit is formed by a 74LS32 (U2), an inverter 74LS04 (U5), and a selection jumper. The main difference between the RAM (HM6116) and ROM chip is in the control of pin 21. Pin 21 must be held at +5 volts for the ROM and this is done through the jumper. For operation of the RAM, this pin is the write enable line. The R/W line from the

Apple is gated with the memory select line (U8, pin 18) and the timing signal Q3 to provide the necessary "write" timing sequence. The selection jumper must be in the RAM position for correct RAM operation. (Because of the exact decoding circuitry used, only the first 1920 bytes of RAM are accessible and usable.)

4.3 DEVICE SELECT ADDRESSING

The decoding for the 16 byte device select memory space is done by U8 (74LS139). The first section of U6 uses the device select line (pin 41) and addresses line A3 (pin 5) to decode the device space into two areas of 8 bytes. The first device area (0 thru 7) is used to access the MC68488 IC. The second device area (8 thru F) is further decoded by U6 into a read-only group and a write-only group. The read-only group enables a tri-state buffer U14, pin 1, so that the SRQ signal can be read. The write-only group is sent to a latch (U12) to control the functions IFC, REN and ATN.

DEVICE MAP

Device Address = 49280+(16*SLOT USED)
Device address + index = Function address

INDEX	READ	WRITE
0	Interrupt Status	Interrupt Mask
1	Command Status	Unused
2	Address Status	Address Mode
3	Auxiliary Command	Auxiliary Command
*4	Address Switch	Address
5	Serial Poll	Serial Poll
6	CMD Pass-Through	Parallel Poll
7	Data In	Data Out
8	SRQ (MSB)	Set ATN True (0)
9	SRQ	Set ATN False (1)
10	SRQ	Set REN True
11	SRQ	Set REN False
12	SRQ	Set IFC True
13	SRQ	Set IFC False
14	SRQ	Unused
15	SRQ	Unused

*Not available on the A488.

4.4 IFC/REN/ATN SIGNALS

Since the MCM68488 does not have the ability to generate the IFC, REN & ATN signals, an on-board latch (U12) is used through software to create these functions. The A488 firmware manages these lines by accessing the device select slot space.

The latch (U12) uses the address line A0 to set the logic state of one of its 8 outputs. The selection of which latch output will be changed is controlled by address lines A1 & A2. A map of these outputs is shown in Section 4.3.

4.5 IEEE 488 BUFFERING

The interface from the A488 card to the IEEE 488 bus is driven by four special transceiver IC's (U13, U15, U16, U17). The transceivers (MC3446) are designed to meet the unique load termination and drive requirements needed on the 488 bus. The receiver section of MC3446 has a hysteresis of about 900 millivolts that reduces its sensitivity to bus noise. The transmitter section can sink up to 48 ma to drive 15 external devices. The bus termination built into the MC3446 IC is 2.4K ohms from the bus to +5V, and 5.0K ohms from the bus to signal ground.

5.0 TROUBLESHOOTING HINTS

5.1 BUS MANAGEMENT LINES ATN, REN & IFC

These lines are controlled by a write to the upper 8 bytes of the Device Select space. A BASIC POKE to (16*slot)+49288 will assert ATN true. This can be read at the interface connector using a logic probe, DVM, or a scope.

POKE (16*slot)+49289,X will assert ATN False. (The value of X doesn't matter!)

POKE (16*slot)+49290,X should force REN True, and POKE(16*slot+49291),X will force REN False.

POKE (16*slot)+49292,X should force IFC True, and POKE (16*slot+49293),X will force IFC False.

If a problem is found in the above section, then check the area around IC's: U12 (74LS259), U16 (3446), and U6 (74LS139). These three IC's provide most of the above control.

5.2 ROM MAPPING ENABLE CIRCUITRY

When operating within the Apple monitor program, type in CN00L (where N = slot number). This will disassemble 20 lines of the ROM code. Compare the first three lines with the listing below. This should also enable the ROM mapping circuit for the expansion area.

```
CN00-  8D FF CF   STA   $CFFF
CN03-  AD F8 07   LDA   $07F8
CN06-  48         PHA
```

Type in C800L. This screen listing should agree with what you saw in the previous step.

If a problem is found in any of the above steps, check the area around IC's: U3(74LS08), U1(74LS132), and U4(74LS138). These IC's make up the Bank Enable section.

5.3 ROM MAPPING DISABLE CIRCUITRY

Perform the steps outlined in Section 5.2.

Now (still in the Apple monitor), type in CFFF. This should disable the expansion ROM.

Type in C800L and verify that the display does not agree with what you saw in Step 5.2.

If a problem is found in this step, the Disable is not working. Check the area of U4(74LS138) and U1(74LS132).

5.4 MC68488 CHIP FUNCTION

Enter the Apple monitor. (In the following steps, "N" is equal to 8 + slot number in Hex.)

Type in C0N3:80; then type in C0N3:00. This should reset the 68488 chip.

Type in C0N2:C0. Then type in C0N2. The Apple should print C0N2-00.

Type in C0N2:C0; then type in C0N2. The Apple should print C0N2-C8. This verifies that the 68488 chip can be addressed and read.

If a problem is found in the above steps, check the area of C6(MC68488), D1(74LS139), D5(74LS08), and D6(74LS245).

6.0 WARRANTY

SSM Microcomputer Products, Inc., warrants its products to be free from defects in materials and/or workmanship for a period of one (1) year for factory assembled boards. In the event of malfunction or other indication of failure attributable directly to faulty workmanship and/or material, then, upon return of the product (shipping prepaid) to SSM at 2190 Paragon Drive, San Jose, California 95131, "Attention: Warranty Claims Department", SSM will, at its option, repair or replace the defective part or parts to restore said product to proper operating condition. All such repairs and/or replacements shall be rendered by SSM without charge for parts or labor when the product is returned within the specified period of the date of purchase. This warranty applies only to the original purchaser.

This warranty will not cover the failure of SSM products which, at the discretion of SSM, shall have resulted from accident, abuse, negligence, alteration, or misapplication of the product. While every effort has been made to provide clear and accurate technical information on the application of SSM products, SSM assumes no liability in any events which may arise from the use of said technical information.

This warranty is in lieu of all other warranties, expressed or implied, including warranties of mercantability and fitness for use. In no event will SSM be liable for incidental and consequential damages arising from or in any way connected with the use of its products. Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

IMPORTANT: Proof of purchase is necessary for products returned for repair under warranty. Before returning any product please call our Customer Service Department for a return authorization number.

A488 PARTS LIST

1	U5	74L$04	hex inverter
1	U3	74L$08	quad 2-input NAND gate
1	U2	74L$32	quad 2-input OR gate
1	U1	74L$132	quad 2-input NAND gate (Schmitt trigger)
1	U4	74L$138	3-to-8 line decoder
1	U6	74L$139	dual 2-to-4 line decoder
1	U9	74L$244	octal buffer
1	U10	74L$245	octal bus transceivers
1	U12	74L$259	8-bit addressable latch
2	U11,14	74L$367	hex bus driver
4	U13, 15-17	MC3446	quad interface bus transceiver
1	U7	MC68488	general purpose interface adapter
1	U8	2716	A488 firmware
1	R1	470 ohm, 1/4 W 5% (yellow, violet, brown)	
1	R2	2.7K ohm, 1/4 W 5% (red, violet, red)	
1	C2	100 pf disc capacitor	
10	C1, 3-11	.1 uf 1/4" spacing monolithic	
4		14 pin sockets	
9		16 pin sockets	
2		20 pin sockets	
1		24 pin socket	
1		40 pin socket	
2		Mini-jumpers	
1		2 pin (wire wrap pin)	
1		3 pin (wire wrap pin)	
1		13x2 wire pin header	
1		72" cable with male/female connector	
1		A488 PC Board	
1		A488 Instruction Manual	
1		Warranty Card	

MOTOROLA Semiconductors

3501 ED BLUESTEIN BLVD., AUSTIN, TEXAS 78721

MC68488

Advance Information

MOS
(N-CHANNEL, SILICON-GATE DEPLETION LOAD)

GENERAL PURPOSE INTERFACE ADAPTER

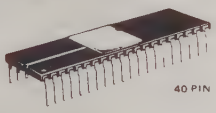
GENERAL PURPOSE INTERFACE ADAPTER

The MC68488 GPIA provides the means to interface between the IEEE488 standard instrument bus and the MC6800 The 488 instrument bus provides a means for controlling and moving data from complex systems of multiple instruments

The MC68488 will automatically handle all handshake protocol needed on the instrument bus

- Single or dual primary address recognition
- Secondary address capability
- Complete source and acceptor handshakes
- Programmable interrupts
- RFD holdoff to prevent data overrun
- Operates with DMA controller
- Serial and parallel polling capability
- Talk-only or listen-only capability
- Selectable automatic features to minimize software
- Synchronization trigger output
- M6800 bus compatible

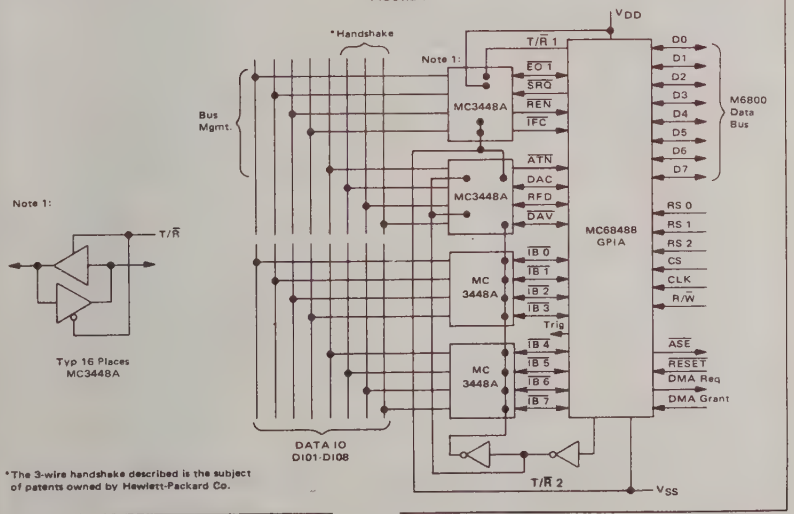
L SUFFIX
CERAMIC PACKAGE
CASE 715



NOT SHOWN

P SUFFIX
PLASTIC PACKAGE
CASE 711

FIGURE 1



*The 3-wire handshake described is the subject of patents owned by Hewlett-Packard Co.

This is advance information and specifications are subject to change without notice

© MOTOROLA INC 1977

ADI-462
(Replaces NP-69)

GENERAL DESCRIPTION

The IEEE 488 instrument bus standard is a bit-parallel, byte-serial bus structure designed for communication to and from intelligent instruments. Using this standard, many instruments may be interconnected and remotely and automatically controlled or programmed. Data may be taken from, sent to, or transferred between instruments. A bus controller dictates the role of each device by making the attention line true and sending talk or listen addresses on the instrument bus data lines; those devices which have matching addresses are activated. Device addresses are set into each GPIA from switches or jumpers on a PC board by a microprocessor as a part of the initialization sequence.

When the controller makes the attention line true, instrument bus commands may also be sent to single or multiple GPIA's.

Information is transmitted on the instrument bus data lines under sequential control of the three handshake lines. No step in the sequence can be initiated until the previous step is completed. Information transfer can proceed as fast as the devices can respond, but no faster than the slowest device presently addressed as active. This permits several devices of different speeds to receive the same data concurrently.

The GPIA is designed to work with standard 488-bus driver IC's (MC3448A's) to meet the complete electrical specifications of the IEEE 488 bus. Additionally, a powered-off instrument may be powered-on without disturbing the 488 bus. With some additional logic, the GPIA could be used with other microprocessors.

The MC68488 GPIA has been designed to interface between the MC6800 microprocessor and the complex protocol of the IEEE 488 instrument bus. Many instrument bus protocol functions are handled automatically by the GPIA and require no additional MPU action. Other functions require minimum MPU response due to a large number of internal registers conveying information on the state of the GPIA and the instrument bus.

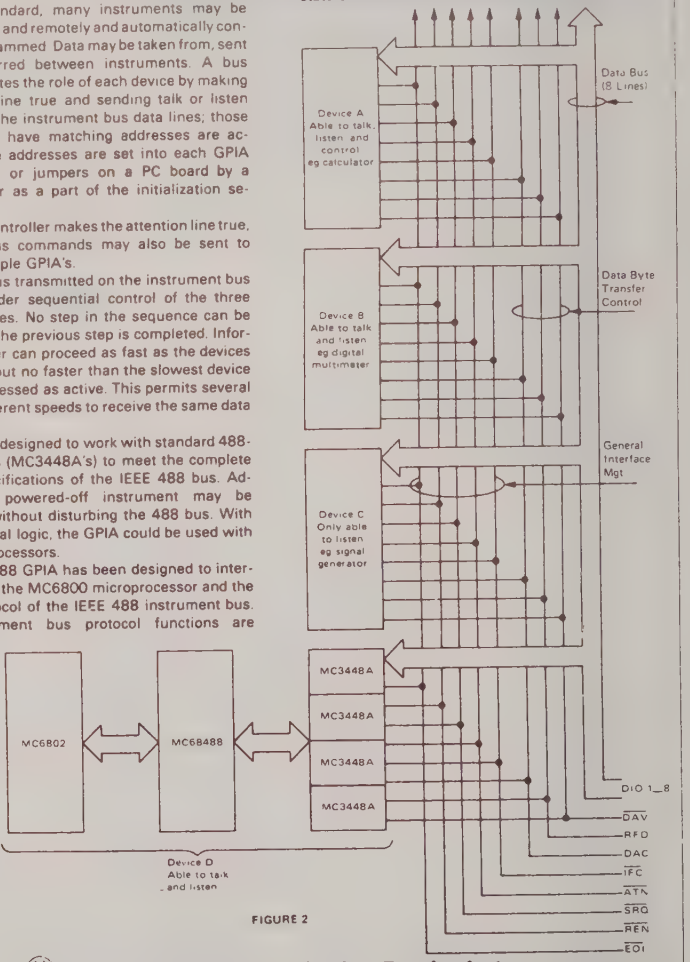


FIGURE 2

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.3 to +7.0	Vdc
Input Voltage	V_{in}	-0.3 to +7.0	Vdc
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	-55 to +150	°C
Thermal Resistance	θ_{JA}	82.5	°C/W

This device contains circitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit

ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 V ±5%, V_{SS} = 0, T_A = 0 to 70°C unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	V_{SS} +2.0	–	V_{CC}	Vdc
Input Low Voltage		V_{IL}	V_{SS} -0.3	–	V_{SS} +0.8	
Input Leakage Current (V_{in} = 0 to 5.25 V)		I_{in}	–	1.0	2.5	μAdc
Three-State (Off State) Input Current (V_{in} = 0.4 to 2.4 V)	D0–D7	I_{TSI}	–	2.0	10	μAdc
Output High Voltage (I_{load} = -205 μA)	D0–D7	V_{OH}	V_{SS} +2.4	–	–	Vdc
Output Low Voltage (I_{load} = 1.6 mA) (I_{load} = 3.2 mA)	D0–D7 $\overline{IRQ}$	V_{OL}	– –	– –	V_{SS} +0.4 V_{SS} +0.4	Vdc
Output Leakage Current (Off State) (V_{OH} = 2.4 Vdc)	$\overline{IRQ}$	I_{LOH}	–	1.0	10	μAdc
Power Dissipation		P_D	–	600	–	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0–D7 All others	C_{in}	– –	– –	12.5 7.5	pF

FIGURE 3 – SOURCE and ACCEPTOR HANDSHAKE

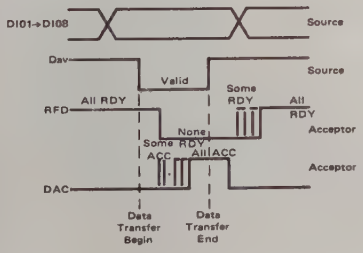
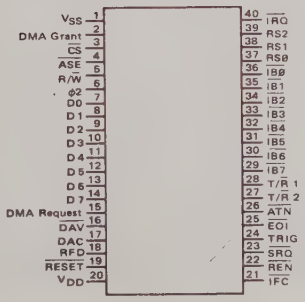


FIGURE 4 – PIN ASSIGNMENT



BUS TIMING CHARACTERISTICS

Characteristic	Symbol	Min	Max	Unit
READ (See Figure 5)				
Enable Cycle Time	t_{cycE}	1.0	–	µs
Enable Pulse Width, High	PW_{EH}	0.45	–	µs
Enable Pulse Width, Low	PW_{EL}	0.43	–	µs
Setup Time, Address and R/$\overline{W}$ valid to enable positive transition	t_{AS}	160	–	ns
Data Delay Time	t_{DDR}	–	320	ns
Data Hold Time	t_H	10	–	ns
Address Hold Time	t_{AH}	10	–	ns
Rise and Fall Time for Enable input	t_{Er}, t_{Ef}	–	25	ns
WRITE (See Figure 6)				
Enable Cycle Time	t_{cycE}	1.0	–	µs
Enable Pulse Width, High	PW_{EH}	0.45	–	µs
Enable Pulse Width, Low	PW_{EL}	0.43	–	µs
Setup Time, Address and R/$\overline{W}$ valid to enable positive transition	t_{AS}	160	–	ns
Data Setup Time	t_{DSW}	195	–	ns
Data Hold Time	t_H	10	–	ns
Address Hold Time	t_{AH}	10	–	ns
Rise and Fall Time for Enable input	t_{Er}, t_{Ef}	–	25	ns
OUTPUT (See Figure 7)				
Output Delay Time $\overline{DAV}$, DAC, RFD, $\overline{EOI}$, $\overline{ATN}$ valid T/$\overline{R}$1, T/$\overline{R}$2 valid	t_{HD} $t_{T/\overline{R}1, 2D}$		400 400	ns ns

FIGURE 5 – BUS READ TIMING CHARACTERISTICS
(Read Information from GPIA)

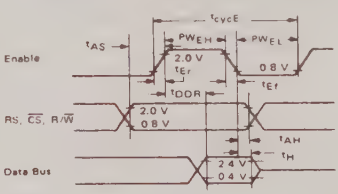


FIGURE 6 – BUS WRITE TIMING CHARACTERISTICS
(Write Information into GPIA)

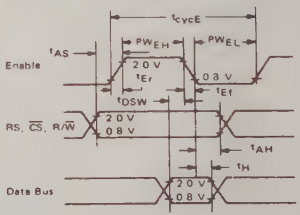
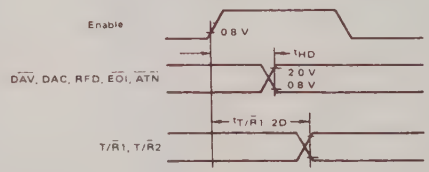


FIGURE 7 – OUTPUT BUS TIMING



AC TIME VALUES

Characteristics		Symbol*	Typ	Unit
Settling Time for Multiple Message	SH	T_1	≥2	μs**
Response to $\overline{ATN}$	SH, AH, T, L	t_2	≤200	ns
Interface Message Accept Time ‡	AH	T_3	>0	§
Response to $\overline{IFC}$ or $\overline{REN}$ False	T, TE, L, LE	t_4	<100	μs
Response to $\overline{ATN}$ • $\overline{EOI}$	PP	t_5	≤200	ns

* Time values specified by a lower case t indicate the maximum time allowed to make a state transition. Time values specified by an upper case T indicate the minimum time that a function must remain in a state before exiting.

** If three-state drivers are used on the $\overline{DIO}$ – $\overline{DAV}$ and $\overline{EOI}$ lines, T_1 may be:
(1) ≥1100 ns
(2) Or ≥700 ns if it is known that within the controller $\overline{ATN}$ is driven by a three-state driver.
(3) Or ≥500 ns for all subsequent bytes following the first sent after each false transition of $\overline{ATN}$ [the first byte must be sent in accordance with (1) or (2)].

‡ Time required for interface functions to accept, not necessarily respond to interface messages.

§ Implementation dependent.

MPU bus clock rate – The current 6800 bus clock is ≤ 1 MHz but part should operate at 1.5 MHz (design goal), with appropriate settling times (T1).

GPIA/MPU INTERFACE SIGNALS

The MC68488 interfaces to the M6800 MPU with an eight-bit bidirectional data bus, a chip select, Read/$\overline{Write}$ line, $\overline{reset}$ line, three register select lines, an interrupt request line, two DMA control lines, and an address switch enable line.

Bidirectional Data (D0-D7)—The bidirectional data lines allow the transfer of data between the MPU and the GPIA. The data bus output drives are three state devices that remain in the high impedance (off) state except when the MPU performs a GPIA read operation. The Read/$\overline{Write}$ line is in the read state when the GPIA is selected for a read operation.

Chip Select ($\overline{CS}$)—This input signal is used to select the GPIA. $\overline{CS}$ must be low for selection of the device. Chip select decoding is normally accomplished with logic external to the chip.

Read/$\overline{Write}$ Line (R/$\overline{W}$)—This signal is generated by the MPU to control register access and direction of data transfer on the data bus. A low state on the GPIA Read/$\overline{Write}$ allows for the selection of one of seven write only registers when used in conjunction with the register select lines; RS0, RS1, RS2. A high state on the GPIA Read/$\overline{Write}$ allows for the selection of one of eight read only registers when used in conjunction with register select lines RS0, RS1, RS2.

Register Select (RS0, RS1, RS2)—The three register select lines are used to select the various registers inside the GPIA. These three lines are used in conjunction with the Read/$\overline{Write}$ line to select a particular register that is to be written or read. Table 1 shows the register select coding.

Interrupt Request ($\overline{IRQ}$)—The $\overline{IRQ}$ output goes to the common interrupt bus for the MPU. This is an open drain output which is wire-ORed to the $\overline{IRQ}$ bus. The $\overline{IRQ}$ is set false (low) when an enabled interrupt occurs and stays false until the MPU reads from the interrupt status register.

$\overline{Reset}$—The active low $\overline{Reset}$ line is used to initialize the chip during power on start up. $\overline{Reset}$ will be driven by an external power-up reset circuit.

DMA Control Lines (DMA Grant, DMA Request)—The DMA request line is used to signal waiting data when Byte In (BI) or Byte Out (BO) is set high for a DMA controller. The DMA request line is set high if either the BI or BO interrupt flags are set in the Interrupt Status Register (ROW) and the corresponding bits in the Interrupt Mask Register (ROR) are set true. The DMA request line is cleared when the DMA grant is made true. The DMA grant line is used to signal the GPIA that the DMA controller has control of the MPU data and address lines. **DMA Grant must be grounded when not in use!*

Address Switch Enable ($\overline{ASE}$)—The $\overline{ASE}$ output is used to enable the device address switch three state buffers to allow the instrument address switch to be read on the MPU bus

Clock Input (Clk)—The Clk input is normally a derivative of the MPU φ2 clock

TABLE 1 — Register Access

RS2	RS1	RS0	R/W	Register Title	Register Symbol
0	0	0	1	Interrupt Status	R0R
0	0	0	0	Interrupt Mask	R0W
0	0	1	1	Command Status	R1R
0	0	1	0	Unused	—
0	1	0	1	Address Status	R2R
0	1	0	0	Address Mode	R2W
0	1	1	1	Auxiliary Command	R3R
0	1	1	0	Auxiliary Command	R3W
1	0	0	1	Address Switch*	R4R
1	0	0	0	Address	R4W
1	0	1	1	Serial Poll	R5R
1	0	1	0	Serial Poll	R5W
1	1	0	1	Command Pass-Through	R6R
1	1	0	0	Parallel Poll	R6W
1	1	1	1	Data In	R7R
1	1	1	0	Data Out	R7W

*External to MC68488

GPIA/488 Interface Bus Signals

The GPIA provides a set of eighteen interface signal lines between the M6800 and the IEEE Standard 488 bus

Signal Lines ($\overline{IB0}$-$\overline{IB7}$)—These bidirectional lines allow for the flow of seven bit ASCII interface messages and device dependent messages. Data appears on these lines in a bit-parallel byte-serial form. These lines are buffered by the MC3448A transceivers and applied to the 488 bus (DIO1-DIO8)

Byte Transfer Lines (DAC, RFD, $\overline{DAV}$)—These lines allow for proper transfer of each data byte on the bus between sources and acceptors. RFD goes passively true indicating that all acceptors are "ready for data." A source will indicate the "data is valid" by pulling $\overline{DAV}$ low. Upon the reception of valid data by all acceptors, DAC will go passively true indicating that the "data has been accepted" by all acceptors

Bus Management Lines ($\overline{ATN}$, $\overline{IFC}$, $\overline{SRQ}$, $\overline{EOI}$, $\overline{REN}$)—These lines are used to manage an orderly flow of information across the interface lines.

Attention ($\overline{ATN}$)—Is sent true over the interface to disable current talker and listeners freeing the signal lines ($\overline{IB0}$-$\overline{IB7}$). During the $\overline{ATN}$ active state devices monitor the DIO1 for addressing or an interface command. Data flows on the DIO1 lines when $\overline{ATN}$ is inactive (high).

Interface Clear ($\overline{IFC}$)—Is used to put the interface system into a known quiescent state.

Service Request ($\overline{SRQ}$)—Is used to indicate a need for attention in addition to requesting an interruption in the current sequence of events. This indicates to the controller that a device on the bus is in need of service.

Remote Enable ($\overline{REN}$)—is used to select one of two alternate sources of device programming data, local or remote control.

End of Identify ($\overline{EOI}$)—is used to signal the end of a multiple byte transfer sequence and in conjunction with $\overline{ATN}$ executes a parallel polling sequence

Transmit/Receive Control Signals (T/$\overline{R}$ 1, T/$\overline{R}$ 2)—These two signals are used to control the quad transceivers which drive the interface bus. It is assumed that transceivers equivalent to the MC3448A will be used where each transceiver has a separate transmit/$\overline{receive}$ control pin. These pins can support one TTL load each. The outputs can then be grouped as shown in Figure 1 with $\overline{SRQ}$ hardwired high to transmit. The transmit/$\overline{receive}$ inputs of $\overline{REN}$, $\overline{IFC}$, and $\overline{ATN}$ are hardwired low to receive. $\overline{EOI}$ is controlled by T/$\overline{R}$1 through the MC3448A (or an equivalent) allowing it to transmit or receive. T/$\overline{R}$1 operates exactly as T/$\overline{R}$2 except during the parallel polling sequence. During parallel poll $\overline{EOI}$ will be made an input by T/$\overline{R}$1 while $\overline{DAV}$ and $\overline{IB0}$/$\overline{IB7}$ lines are outputs.

GPIA INTERNAL CONTROLS AND REGISTERS*

There are fifteen locations accessible to the MPU data bus which are used for transferring data to control the various functions on the chip and provide current chip status. Seven of these registers are write only and eight registers are read only. The various registers are accessed according to the three least significant bits of the MPU address bus and the status of the Read/Write line. One of the

*NOTE Upper and lower case type designations will be used with the register bits to indicate remote or local messages respectively

fifteen registers is external to the IC but an address switch register is provided for reading the address switches. Table 2 shows actual bit contents of each of the registers.

Data-In Register R7R—The data-in register is an actual eight-bit storage register used to move data from the interface bus when the chip is a listener. Reading the register does not destroy information in the data-out register. DAC (data accepted) will remain low until the MPU removes the byte from the data-in register. The chip will automatically finish the handshake by allowing DAC to go high. In RFD (ready for data) holdoff mode, a new handshake is not initiated until a command is sent allowing the chip to release holdoff. This will delay a talker until the available information has been processed.

DATA IN REGISTER
(Read Only)

DI7	DI6	DI5	DI4	DI3	DI2	DI1	DIØ

DIØ-DI7 – Correspond to DIO1-DIO8 of the 488 1975 Standard and $\overline{IBØ}$-$\overline{IB7}$ of the MC68488

Data-Out Register R7W — The data-out register is an actual eight-bit storage register used to move data out of the chip onto the interface bus Reading from the data-in register has no effect on the information in the data-out register. Writing to the data-out register has no effect on the information in the data-in register.

DATA OUT REGISTER
(Write Only)

DO7	DO6	DO5	DO4	DO3	DO2	DO1	DOØ

DOØ-DO7 – Correspond to DIO1-DIO8 of the 488 1975 Standard and $\overline{IBØ}$-$\overline{IB7}$ of the MC68488

Interrupt Mask Register R0W — The Interrupt Mask Register is a seven-bit storage register used to select the particular events that will cause an interrupt to be sent to the MPU. The seven control bits may be set independently of each other. If dsel (bit 7 of the Address Mode Register) is set high CMD bit 2 will interrupt on SPAS or RLC. If dsel is set low CMD will interrupt on UACG, UUCG, and DCAS in addition to RLC and SPAS. The Command Status

TABLE 2

	7	6	5	4	3	2	1	0	
R0W	IRQ	BO	GET		APT	CMD	END	BI	Interrupt "Mask" Reg
R0R	INT	BO	GET		APT	CMD	END	BI	Interrupt Status Reg
R1R	UACG	REM	LOK		RLC	SPAS	DCAS	UUCG	Command Status Reg
R1W									Unused
R2R	ma	to	lo	ATN	TACS	LACS	LPAS	TPAS	Address Status Reg
R2W	dsel	to	lo		hide	hlda		apte	Address Mode Reg
R3R	Reset	DAC	$\overline{DAV}$	RFD	msa	rtl	ulpa	fget	Auxiliary Command Reg
R3W		rfdr	feoi	dacr			dacd		
R4R	UD3	UD2	UD1	AD5	AD4	AD3	AD2	AD1	Address Switch Reg
R4W	lsbe	dal	dat	AD5	AD4	AD3	AD2	AD1	Address Register
R5R	S8	SRQS	S6	S5	S4	S3	S2	S1	Serial Poll Reg
R5W		rsv							
R6R	B7	B6	B5	B4	B3	B2	B1	B0	Command Pass thru Reg
R6W	PPR8	PPR7	PPR6	PPR5	PPR4	PPR3	PPR2	PPR1	Parallel Poll Reg.
R7R	D17	D16	D15	D14	D13	D12	D11	DI0	Data In Register
R7W	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0	Data Out Register

Register R1R may then be used to determine which command caused the interrupt. Setting GET bit 5 allows an interrupt to occur on Group Execute Trigger Command. END bit 1 allows an interrupt to occur if $\overline{EOI}$ is true (low) and $\overline{ATN}$ is false (high). APT bit 3 allows an interrupt to occur indicating that a secondary address is available to be examined by the MPU if apte (bit 0 of Address Mode Register) is enabled and listener or talker primary address is received and a Secondary Command Group is received. A typical response for a valid secondary address would be to set msa (bit 3 of Auxiliary Command Register) true and dacr (bit 4 Auxiliary Command Register) true, releasing the DAC handshake. BI indicates that a data byte is waiting in the data-in register. BI is set high when data-in register is full. BO indicates that a byte from the data-out register has been accepted. BO is set when the data-out register is empty. IRQ enabled high allows any interrupt to be passed to the MPU.

INTERRUPT MASK REGISTER

(Write Only)

IRQ	BO	GET	X	APT	CMD	END	BI

IRQ – Mask bit for IRQ pin
BO – Interrupt on byte output
GET – Interrupt on Group Execute Trigger
APT – Interrupt on Secondary Address Pass-Through
CMD – Interrupt on SPAS + RLC + dsel (DCAS + UUCG + UACG)
END – Interrupt on EOI and $\overline{ATN}$
BI – Interrupt on byte input

The Interrupt Status Register R0R — The Interrupt Status Register is a seven-bit storage register which corresponds to the interrupt mask register with an additional bit INT bit 7. Except for the INT bit the other bits in the status register are set regardless of the state of the interrupt mask register when the corresponding event occure. The $\overline{IRQ}$ (MPU interrupt) is cleared when the MPU reads from the register. INT bit 7 is the logical OR of the other six bits ANDed with the respective bit of R0W.

INTERRUPT STATUS REGISTER

(Read Only)

INT	BO	GET	X	APT	CMD	END	BI

INT – Logical OR of all other bits in this register ANDed with the respective bits in the interrupt mask register.
BO – A byte of data has been output
GET – A Group Execute Trigger has occurred
APT – An Address Pass Through has occurred
CMD – SPAS + RLC + dsel (DCAS + UUCG + UACG) has occurred
END – An EOI has occurred with $\overline{ATN}$ = 0
BI – A byte has been received

Serial Poll Register R5R/W — The Serial Poll Register is an eight-bit storage register which can be both written into and read by the MPU. It is used for establishing the status byte that the chip sends out when it is serial poll enabled. Status may be placed in bits 0 through 5 and bit 7. Bit 6 rsv (request for service) is used to drive the logic which controls the $\overline{SRQ}$ line on the bus telling the controller that service is needed. This same logic generated the signal SRQS which is substituted in bit 6 position when the status byte is read by the MPU $\overline{IB0}$–$\overline{IB7}$. In order to initiate a rsv (request for service), the MPU sets bit 6 true (generating rsv signal) and this in turn causes the chip to pull down the $\overline{SRQ}$ line. SRQS is the same as rsv when SPAS is false. Bit 6 as read by the MPU will be the SRQS (Service Request State).

SERIAL POLL REGISTER

(Read)

S8	SRQS	S6	S5	S4	S3	S2	S1

S1-S8 – Status bits
SRQS – Bus in Service Request State

SERIAL POLL REGISTER

(Write)

S8	rsv	S6	S5	S4	S3	S2	S1

S1-S8 – Status bits
rsv – generate a service request

Parallel Poll Register R6W—This register will be loaded by the MPU and the bits in this register will be delivered to the instrument bus IB0–IB7 during PPAS (Parallel Poll Active State). This register powers up in the PPO (Parallel Poll No Capability) state. The reset bit (Auxiliary Command Register bit 7) will clear this register to the PPO state.

The parallel poll interface function is executed by this chip using the PP2 subset (Omit Controller Configuration Capability). The controller cannot directly configure the parallel poll output of this chip. This must be done by the MPU. The controller will be able to indirectly configure the parallel poll by issuing an addressed command which has been defined in the MPU software.

PARALLEL POLL REGISTER

(Write Only)

PP8	PP7	PP6	PP5	PP4	PP3	PP2	PP1

Bits delivered to bus during Parallel-Poll Active State (PPAS)
Register powers up in the PP0 state
Parallel Poll is executed using the PP2 subset

Address Mode Register R2W — The address mode register is a storage register with six bits for control: to, lo, hide, hlda, dsel, and apte. The to bit 6 selects the talker/listener and addresses the chip to talk only. The lo bit 5 selects the talker/listener and sets the chip to listen only. The apte bit 0 is used to enable the extended addressing mode. If apte is set low the device goes from the TPAS (Talker Primary

Address State) directly to the TADS (Talker Addressed State). The hlda bit 2 holds off RFD (Ready for Data) on ALL DATA until rfdr is set true. The hlde bit 3 holds off RFD on EOI enabled (low) and ATN not enabled (high). This allows the last byte in a block of data to be continually read as needed. Writing rfdr true (high) will allow the next handshake to proceed.

ADDRESS MODE REGISTER
(Write Only)

dsel	to	lo	X	hdle	hdla	X	apte

dsel - configure for automatic completion of handshake sequence on occurrence of GET, UACG, UUCG, SDC, or DCL commands
to - set to talk only mode
lo - set to listen only mode
hdle - Hold off RFD on end
hdla - Hold off RFD on all data
apte - Enable the address pass through feature

Address Status Register R2R — The address status register is not a storage register but simply an eight-bit port used to couple internal signal nodes to the MPU bus. The status flags represented here are stored internally in the logic of the chip. These status bits indicate the addressed state of the talker/listener as well as flags that specify whether the chip is in the talk only or listen only mode. The ATN, bit 4, contains the condition of the Attention Line. The ma signal is true when the chip is in:

TACS — Talker Active State
TADS — Talker Addressed State
LACS — Listener Active State
LADS — Listener Addressed State
SPAS — Serial Poll Active State

ADDRESS STATUS REGISTER
(Read Only)

ma	to	lo	ATN	TACS	LACS	LPAS	TPAS

ma - my address has occurred
to - the talk only mode is enabled
lo - the listen only mode is enabled
ATN - The Attention command is asserted
TACS - GPIA is in the Talker Active State
LACS - GPIA is in the Listener Active State
LPAS - GPIA is in the Listener Primary Addressed State
TPAS - GPIA is in the Talker Primary Addressed State

Address Switch Register R4R — The address switch register is external to the chip. There is an enable line ($\overline{ASE}$) to be used to enable three-state drivers connected between the address switches and the MPU. When the MPU addresses the address switch register the enable line directs the switch information to be sent to the MPU. The five least significant bits of the eight-bit register are used to specify the bus address of the device and the remaining three bits may be used at the discretion of the user. The most probable use of one or two of the bits is for controlling the listener only or talk only functions.

ADDRESS SWITCH REGISTER
(Read Only)

UD3	UD2	UD1	AD5	AD4	AD3	AD2	AD1

AD1-AD5 - Device address
UD1-UD3 - User definable bits
When this "register" is addressed, the $\overline{ASE}$ pin is set which allows external address switch information from bus device to be read

Address Register R4W — The Address Register is an eight-bit storage register. The purpose of this register is to carry the primary address of the device. The primary address is placed in the five least significant bits of the register. If external switches are used for device addressing these are normally read from the Address Switch Register and then placed in the Address Register by the MPU.

AD1 through AD5 bits 0–5 are for the device's address. The lsbe bit 7 is set to enable the Dual Primary Addressing Mode. During this mode the device will respond to two consecutive addresses, one address with AD1 equal to 0 and the other address with AD1 equal to 1. For example, if the device's address is HEX 0F, the Dual Primary Addressing Mode would allow the device to be addressed at both HEX OF and HEX OE. The dal bit 6 is set to disable the listener and the dat bit 5 is set to disable the talker.

This register is cleared by the $\overline{Reset}$ input only (not by the reset bit of the Auxiliary Command Register bit 7).

When $\overline{ATN}$ is enabled and the primary address is received on the $\overline{IBO-7}$ lines, the MC68488 will set bit 7 of the address status register (ma). This places the MC68488 in the TPAS or LPAS.

When $\overline{ATN}$ is disabled the GPIA may go to one of three states: TACS, LACS or SPAS.

ADDRESS REGISTER
(Write Only)

lsbe	dal	dat	AD5	AD4	AD3	AD2	AD1

lsbe – enable dual primary addressing mode
dal – disable the listener
dat – disable the talker
AD1-AD5 . Primary device address, usually read from address switch register
Register is cleared by the $\overline{Reset}$ input pin only

Auxiliary Command Register R3R/W — Bit 7, reset, initializes the chip to the following states: (reset is set true by external $\overline{Reset}$ input pin and by writing into the register from the MPU).

SIDS—Source Idle State
AIDS—Acceptor Idle State
TIDS—Talker Idle State
LIDS—Listener Idle State

LOCS—Local State
NPRS—Negative Poll Response State
PPIS—Parallel Poll Idle State
PUCS—Parallel Poll Unaddressed to Configure State
PPO—Parallel Poll No Capability

rfdr (release RFD handshake) bit 6 allows for completion of the handshake that was stopped by RFD (Ready For Data) holdoff commands hlda and hlde

fget (force group execute trigger) bit 0 has the same effect as the GET (Group Execute Trigger) command from the controller

rtl (return to local) bit 2 allows the device to respond to local controls and the associated device functions are operative.

dacr (release DAC handshake) bit 4 is set high to allow DAC to go passively true. This bit is set to indicate that the MPU has examined a secondary address or an undefined command.

ulpa (upper/lower primary address) bit 1 will indicate the state of the LSB of the address received on the DIO1-8 bus lines at the time the last Primary Address was received. This bit can be read but not written by the MPU.

msa (valid secondary address) bit 3 is set true (high) when TPAS (Talker Primary Addressed State) or LPAS (Listener Primary Addressed State) is true. The chip will become addressed to listen or talk. The primary address must have been previously received

RFD, $\overline{DAV}$, DAC—(Ready For Data, Data Valid, Data Accepted) bits assume the same state as the corresponding signal on the MC68488 package pins. The MPU may only read these bits. These signals are not synchronized with the MPU clock.

dacd (data accept disable) bit 1 set high by the MPU will prevent completion of the automatic handshake on Addresses or Commands. dacr is used to complete the handshake.

feoi (forced end or identify) bit 5 tells the chip to send $\overline{EOI}$ low. The $\overline{EOI}$ line is then returned high after the next byte is transmitted. NOTE: The following signals are not stored but revert to a false (low) level one clock cycle (MPUϕ2) after they are set true (high).

1. rfdr
2. feoi
3. dacr

These signals can be written but not read by the MPU.

AUXILIARY COMMAND REGISTER

	reset	rfdr	feoi	dacr	msa	rtl	dacd	fget	Write
		DAC	DAV	RFD			ulpa		Read

reset – initialize the chip to the following status
(1) all interrupts cleared
(2) following bus states are in effect SIDS, AIDS, TIDS, LIDS, LOCS, PPIS, PUCS, and PPØ
(3) bit is set by Reset input pin
msa – if GPIA is in LPAS or TDAS, setting msa will force GPIA to LADS or TADS
rtl – return to local if local lockout is disabled
ulpa – state of LSB of bus at last primary address receive time
fget – Force group execute trigger command from the MPU has occurred
rfdr – continue handshake stopped by RFD holdoff
feoi – set EOI true, clears after next byte transmitted
dacr – MPU has examined an undefined command or secondary address
dacd – prevents completion of automatic handshake on Addresses or Commands

Command Status Register R1R — The command status register flags commands or states as they occur. These flags or states are simply coupled onto the MPU bus. There are five major address commands. REM shows the remote/local state of the talker/listener. RLC bit 3 is set when a change of state of the remote/local flip-flop occurs and reset when the command status register is read. DCAS bit 1 indicates that either the device clear or selected device clear has been received activating the device clear function. SPAS bit 2 indicates that the SPE command has been received activating the device serial poll function. UACG bit 7 indicates that an undefined address command has been received and depending on programming the MPU decides whether to execute or ignore it. UUCG bit 0 indicates that an undefined universal command has been received.

COMMAND STATUS REGISTER
(Read)

UACG	REM	LOK	X	RLC	SPAS	DCAS	UUCG

UACG - Undefined Addressed Command
REM - Remote Enabled
LOK - Local Lockout Enabled
RLC - Remote/Local State Changed
SPAS - Serial Poll Active State is in effect
DCAS - Device Clear Active State is in effect
UUCG - Undefined Universal Command

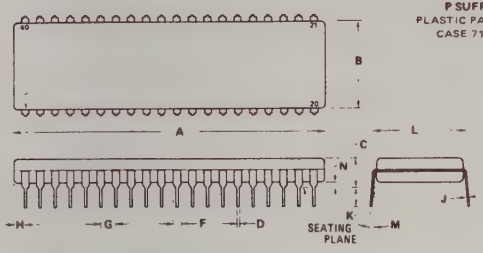
Command Pass-Through Register R6R — The command pass through is an eight-bit port with no storage. When this port is addressed by MPU it

connects the instrument data bus ($\overline{IB0}$–$\overline{IB7}$) to the MPU data bus D0–D7. This port can be used to pass commands and secondary addresses that aren't automatically interpreted through to the MPU for inspection.

COMMAND PASS THROUGH REGISTER
(Read Only)

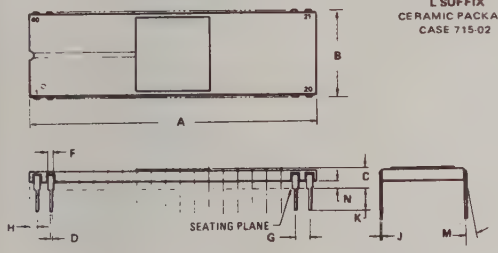
B7	B6	B5	B4	B3	B2	B1	B0

An eight bit input port used to pass commands and secondary addresses to MPU which are not automatically interpreted by the GPIA



P SUFFIX
PLASTIC PACKAGE
CASE 711-01

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	51.82	52.32	2.040	2.060
B	13.72	14.22	0.540	0.560
C	4.57	5.08	0.180	0.200
D	0.36	0.51	0.014	0.020
F	1.02	1.52	0.040	0.060
G	2.41	2.67	0.095	0.105
H	1.65	2.16	0.065	0.085
J	0.20	0.30	0.008	0.012
K	3.68	4.19	0.145	0.165
L	14.99	15.49	0.590	0.610
M	0°	10°	0°	10°
N	0.51	1.02	0.020	0.040



L SUFFIX
CERAMIC PACKAGE
CASE 715-02

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	50.29	51.31	1.980	2.020
B	14.86	15.62	0.585	0.615
C	2.54	4.19	0.100	0.165
D	0.38	0.53	0.015	0.021
F	0.76	1.40	0.030	0.055
G	2.54 BSC		0.100 BSC	
H	0.76	1.78	0.030	0.070
J	0.20	0.33	0.008	0.013
K	2.54	4.19	0.100	0.165
M	0°	10°	0°	10°
N	0.51	1.52	0.020	0.060

NOTE.
1. LEADS, TRUE POSITIONED WITHIN 0.25 mm (0.010) DIA (AT SEATING PLANE), AT MAX. MAT'L CONDITION.

Circuit diagrams utilizing Motorola products are included as a means of illustrating typical semiconductor applications; consequently, complete information sufficient for construction purposes is not necessarily given. The information has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Furthermore, such information does not convey to the purchaser of the semiconductor devices described any license under the patent rights of Motorola Inc. or others.

MOTOROLA

SEMICONDUCTORS

12 • PHOENIX, ARIZONA 85036

MC3446A

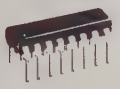
QUAD GENERAL-PURPOSE INTERFACE BUS (GPIB) TRANSCEIVER

The MC3446A is a quad bus transceiver intended for usage in instruments and programmable calculators equipped for interconnection into complete measurement systems. This transceiver allows the bidirectional flow of digital data and commands between the various instruments. The transceiver provides four open-collector drivers and four receivers featuring hysteresis.

- Tailored to Meet the IEEE Standard 488-1978 (Digital Interface for Programmable Instrumentation) and the Proposed IEC Standard on Instrument Interface
- Provides Electrical Compatibility with General-Purpose Interface Bus (GPIB)
- MOS Compatible with High Impedance Inputs
- Driver Output Guaranteed Off During Power Up/Power Down
- Low Power — Average Power Supply Current = 12 mA
- Terminations Provided

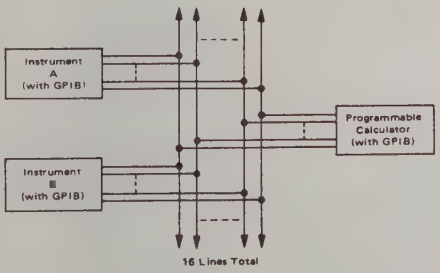
QUAD INTERFACE BUS TRANSCEIVER

SILICON MONOLITHIC INTEGRATED CIRCUIT

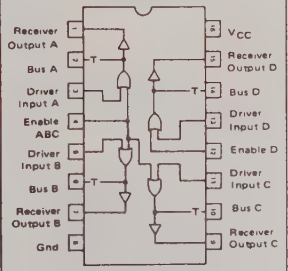


P SUFFIX
PLASTIC PACKAGE
CASE 648

TYPICAL MEASUREMENT SYSTEM APPLICATION



PIN CONNECTIONS



©MOTOROLA INC., 1979

DS 9345 R1

MAXIMUM RATINGS (T_A = 25°C unless otherwise noted.)

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	7.0	Vdc
Input Voltage	V_I	5.5	Vdc
Driver Output Current	$I_{O(D)}$	150	mA
Junction Temperature	T_J	150	°C
Operating Ambient Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C

ELECTRICAL CHARACTERISTICS

(Unless otherwise noted, 4.5 V ≤ V_{CC} ≤ 5.5 V and 0 ≤ T_A ≤ 70°C, typical values are at T_A = 25°C, V_{CC} = 5.0 V.)

Characteristic	Symbol	Min	Typ	Max	Unit
DRIVER PORTION					
Input Voltage – High Logic State	$V_{IH(D)}$	2.0	–	–	V
Input Voltage – Low Logic State	$V_{IL(D)}$	–	–	0.8	V
Input Current – High Logic State (V_{IH} = 2.4 V)	$I_{IH(D)}$	–	5.0	40	µA
Input Current – Low Logic State (V_{IL} = 0.4 V, V_{CC} = 5.0 V, T_A = 25°C)	$I_{IL(D)}$	–	-0.2	-0.25	mA
Input Clamp Voltage (I_{IK} = -12 mA)	$V_{IK(D)}$	–		-1.5	V
Output Voltage – High Logic State (1) ($V_{IH(S)}$ = 2.4 V or $V_{IH(D)}$ = 2.0 V)	$V_{OH(D)}$	2.5	3.3	3.7	V
Output Voltage – Low Logic State ($V_{IL(S)}$ = 0.8 V, $V_{IL(D)}$ = 0.8 V, $I_{OL(D)}$ = 48 mA)	$V_{OL(D)}$	–	–	0.5	
Input Breakdown Current ($V_{I(D)}$ = 5.5 V)	$I_{IB(D)}$	–	–	1.0	mA
RECEIVER PORTION					
Input Hysteresis	–	400	625	–	mV
Input Threshold Voltage – Low to High Output Logic State	$V_{ILH(R)}$	–	1.66	2.0	V
Input Threshold Voltage – High to Low Output Logic State	$V_{IHL(R)}$	0.8	1.03	–	V
Output Voltage – High Logic State ($V_{IH(R)}$ = 2.0 V, $I_{OH(R)}$ = -400 µA)	$V_{OH(R)}$	2.4	–	–	V
Output Voltage – Low Logic State ($V_{IL(R)}$ = 0.8 V, $I_{OL(R)}$ = 8.0 mA)	$V_{OL(R)}$		–	0.5	V
Output Short Circuit Current ($V_{IH(R)}$ = 2.0 V) (Only one output may be shorted at a time)	$I_{OS(R)}$	4.0	–	14	mA
BUS LOAD CHARACTERISTICS					
Bus Voltage ($V_{IH(E)}$ = 2.4 V, I_{BUS} = -12 mA)	$V_{(BUS)}$	2.5 -	3.3 –	3.7 -1.5	V
Bus Current ($V_{IH(D)}$ = 2.4 V, V_{BUS} = 5.0 V) ($V_{IH(D)}$ = 2.4 V, V_{BUS} = 0.5 V) (V_{BUS} = 5.5 V) (V_{CC} = 0.0 V, V_{BUS} = 2.75 V)	$I_{(BUS)}$	0.7 -1.3 - –	– – – –	- -3.2 2.5 0.04	mA
TOTAL DEVICE POWER CONSUMPTION					
Power Supply Current (All Drivers OFF) (All Drivers ON)	I_{CC}	– –	12 32	19 40	mA

SWITCHING CHARACTERISTICS (V_{CC} = 5.0 V, T_A = 25°C)

Characteristic	Symbol	Min	Typ	Max	Unit
DRIVER PORTION					
Propagation Delay Time from Driver Input to Low Logic State Bus Output	$t_{PHL(D)}$	–	–	50	ns
Propagation Delay Time from Driver Input to High Logic State Bus Output	$t_{PLH(D)}$	–	–	40	ns
Propagation Delay Time from Enable Input to Low Logic State Bus Output	$t_{PHL(E)}$	–	-	50	ns
Propagation Delay Time from Enable Input to High Logic State Bus Output	$t_{PLH(E)}$	–	–	50	ns
RECEIVER PORTION					
Propagation Delay Time from Bus Input to High Logic State Receiver Output	$t_{PLH(R)}$	–	–	50	ns
Propagation Delay Time from Bus Input to Low Logic State Receiver Output	$t_{PHL(R)}$	–	–	40	ns

FIGURE 1 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM RECEIVER INPUT (BUS) TO OUTPUT

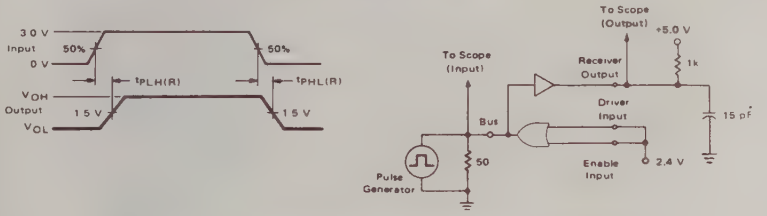
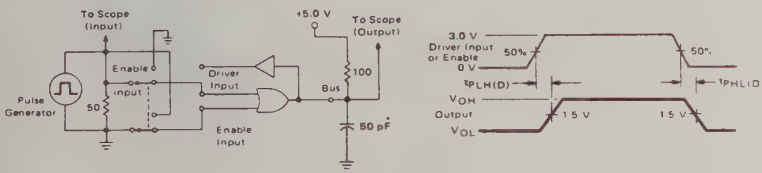


FIGURE 2 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM DRIVER AND COMMON ENABLE INPUTS TO OUTPUT (BUS)



* Includes Probe and Jig Capacitance

FIGURE 3 – TYPICAL RECEIVER HYSTERESIS CHARACTERISTICS

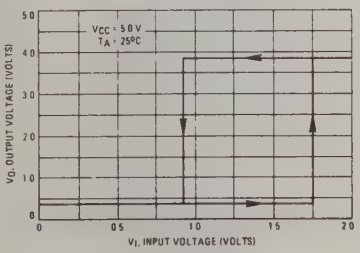
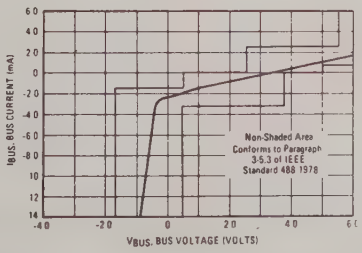
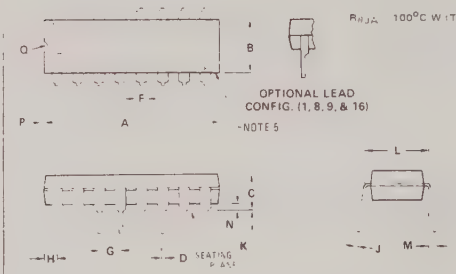


FIGURE 4 – TYPICAL BUS LOAD LINE



OUTLINE DIMENSIONS

P SUFFIX
PLASTIC PACKAGE
CASE 648-05



DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	18.80	21.34	0.740	0.840
B	6.10	6.60	0.240	0.260
C	4.06	5.08	0.160	0.200
D	0.38	0.53	0.015	0.021
F	1.02	1.78	0.040	0.070
G	2.54 BSC		0.100 BSC	
H	0.38	2.41	0.015	0.095
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	7.62 BSC		0.300 BSC	
M	0°	10°	0°	10°
N	0.51	1.02	0.020	0.040

NOTES
1. LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION
2. DIMENSION "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION "B" DOES NOT INCLUDE MOLD FLASH.
4. "F" DIMENSION IS FOR FULL LEADS. "HALF" LEADS ARE OPTIONAL AT LEAD POSITIONS 1, 8, 9, and 16).
5. ROUNDED CORNERS OPTIONAL

THERMAL INFORMATION

The maximum power consumption an integrated circuit can tolerate at a given operating ambient temperature, can be found from the equation

$$P_{D(T_A)} = \frac{T_{J(max)} - T_A}{R_{\theta JA(Typ)}}$$

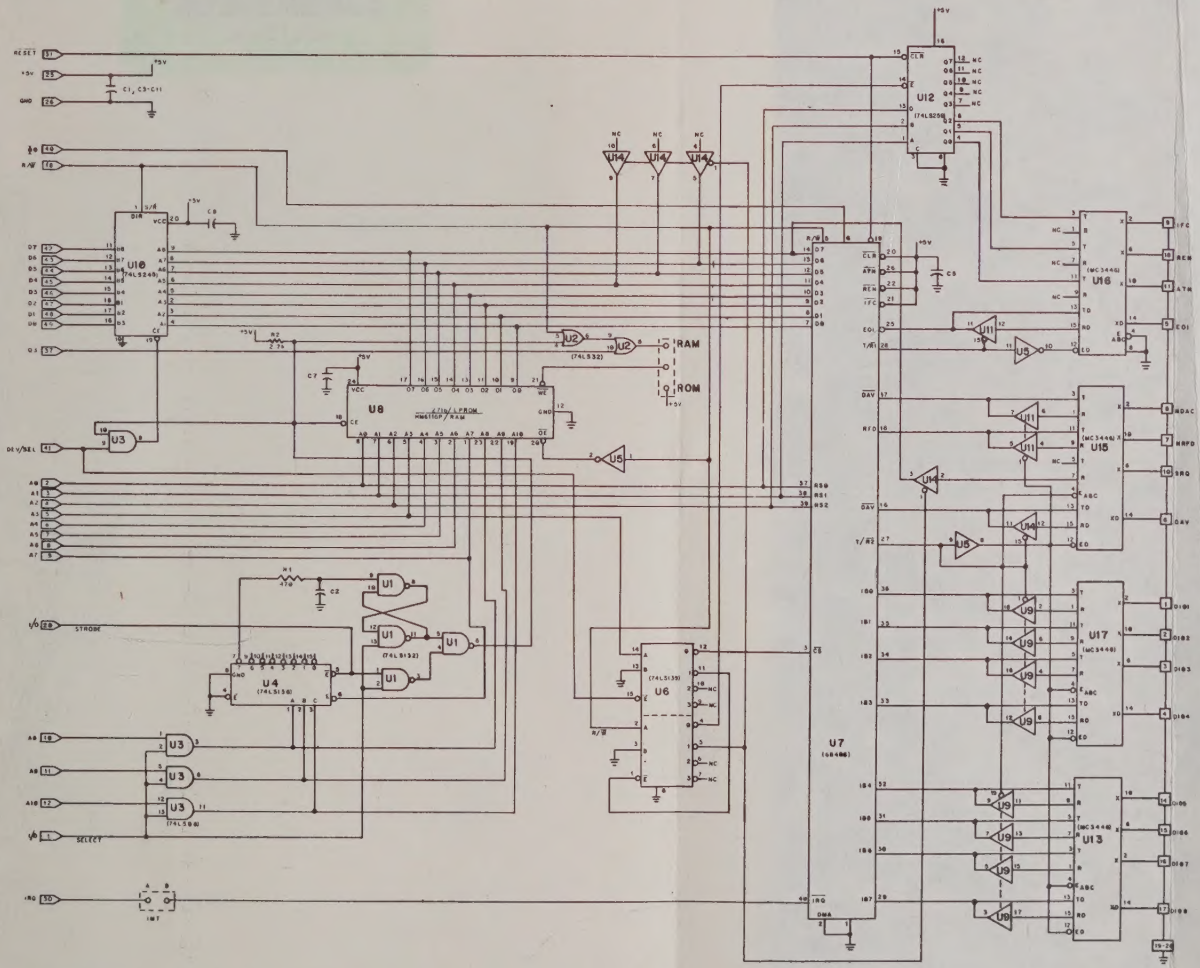
Where, $P_{D(T_A)}$ = Power Dissipation allowable at a given operating ambient temperature. This must be greater than the sum of the products of the supply voltages and supply currents at the worst case operating condition.

$T_{J(max)}$ = Maximum Operating Junction Temperature as listed in the Maximum Ratings Section

T_A = Maximum Desired Operating Ambient Temperature

$R_{\theta JA(Typ)}$ = Typical Thermal Resistance Junction to Ambient

Motorola reserves the right to make changes to any products herein to improve reliability, function or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others.



REV	DESCRIPTION	DATE
A	PRODUCTION RELEASE	12-16-80

NOTES:
1. PINS 18-24 ARE SIGNAL GROUND RETURNS FOR THE REFERENCE SIGNAL.
2. ▱ : BUS CONNECTIONS.
3. ▭ : EXTERNAL BOARD CONNECTIONS.

SSM MICROCOMPUTER PRODUCTS, INC.
© 1980 SSM MICROCOMPUTER PRODUCTS, INC.

SCALE: NONE
DATE: 12-16-80
DRAWN BY: Judy Fisher

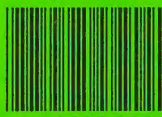
A488-GPIB BOARD SCHEMATIC

SHEET 1 OF 1

1002-002

MP MASTER
MA31556

MP MASTER



MA31556

SSM MICROCOMPUTER PRODUCTS, INC.
2190 Paragon Drive
San Jose, California 95131

(408) 946-7400 • Telex: 171171
TWX: 910-338-2077

ELECTRONIC BULLETIN BOARD: (408) 946-3616